

AN INVESTIGATION INTO A HYBRID GENETIC PROGRAMMING AND
ANT COLONY OPTIMIZATION METHOD FOR CREDIT SCORING

Rojin Aliehyaei

Columbus State University
The College of Business and Computer Science
The Graduate Program in Applied Computer Science

**An Investigation into a Hybrid Genetic Programming and
Ant Colony Optimization Method for Credit Scoring**

A Thesis in
Applied Computer Science

by
Rojin Aliehyaei

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2012

© 2012 by Rojin Aliehyaei

I have submitted this thesis in partial fulfillment of the requirements for the degree of Master of Science.

5/7/2012

Date

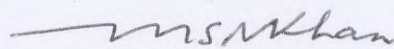


Rojin Aliehyaei

We approve the thesis of Rojin Aliehyaei as presented here:

5/7/2012

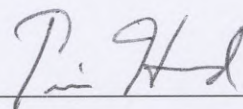
Date



Shamim Khan, Ph.D.
Professor of Computer Science
Thesis Advisor

5/7/2012

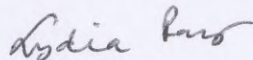
Date



Timothy G. Howard, Ph.D.
Professor of Mathematics

5/7/2012

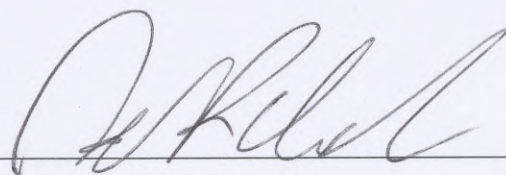
Date



Lydia Ray, Ph.D.
Associate Professor of Computer Science

5/7/2012

Date



Radhouane Chouchane, Ph.D.
Assistant Professor of Computer Science

Abstract

This thesis proposes and investigates a new hybrid technique based on Genetic Programming (GP) and Ant Colony Optimization (ACO) techniques for inducing data classification rules. The proposed hybrid approach aims to improve on the accuracy of data classification rules produced by the original GP technique, which uses randomly generated initial populations. This hybrid technique relies on the ACO technique to produce the initial populations for the GP technique. To evaluate and compare their effectiveness in producing good data classification rules, GP, ACO, and hybrid techniques were implemented in the C programming language. The data classification rules were created and evaluated by executing these codes with two datasets for credit scoring problems, widely known as the Australian and German datasets, available from the Machine Learning Repository at the University of California, Irvine. The experimental results demonstrate that although all tree techniques yield similar accuracy during testing, on average, the hybrid ACO-GP approach performs better than either GP or ACO during training.

Acknowledgements

First, I would like to thank Professor Shamim Khan, who has been an excellent thesis advisor and mentor. Without his guidance, I would not have been able to accomplish several achievements during my time at Columbus State University (CSU). I would also like to thank the School of Computer Science at CSU for supporting my Masters study through a graduate assistantship. Without such financial support, this endeavor would not have been possible.

I am forever grateful to my parents, Abdolali and Mahroo Aliehyaei, for always believing in me and for encouraging me to further my education. I am also deeply thankful to my big sister, Ladan, for always being there for me every day and every night. Finally, I would like to thank my brother and sister, Farid and Yasaman, for their inspirations. I would not be who I am today without my family.

Table of Contents

Abstract.....	iii
Acknowledgements	iv
List of Figures.....	vi
List of Tables	viii
1 Introduction.....	1
2 Related Work	5
3 Proposed Hybrid Technique	11
4 Results and Discussions	21
4.1 The Redesigned ACO Results	21
4.2 The GP Results	28
4.3 The Hybrid Technique Results	32
5 Conclusions and Future Work.....	41
References	43

List of Figures

1 ACO network diagram with attributes from the sample dataset.....	12
2 A sample sub-tree.....	17
3 An example of crossover operation	19
4 Accumulative accuracy of ACO-based rules during training for three experiments on the German dataset	22
5 Accumulative accuracy of ACO-based rules during training for three experiments on the German dataset	23
6 Accumulative accuracy of ACO-based rules on the German dataset during testing.....	24
7 Comparative performance of classification rules extracted by our ACO-based technique against AntMiner+ based on the total number of operators	26
8 Accumulative accuracy of ACO-based rules during training for three experiments on the Australian dataset	27
9 Accumulative accuracy of ACO-based rules during testing for three experiments on the Australian dataset	28
10 Experimental results of the GP technique on the German dataset (training).....	29
11 Experimental results of the GP technique on the German dataset (testing)	30
12 Experimental results of the GP technique on the Australian dataset (training)	31
13 Experimental results of the GP technique on the Australian dataset (testing)	32
14 Experimental results of the hybrid technique with the Big Trees configuration on the German dataset (training).....	33

15 Experimental results of the hybrid technique with the Big Trees configuration on the German dataset (testing)	34
16 Experimental results of the hybrid technique with the Small Trees configuration on the German dataset (training).....	35
17 Experimental results of the hybrid technique with the Small Trees configuration on the German dataset (testing)	36
18 Experimental results of the hybrid technique with the Big Trees configuration on the Australian dataset (training).....	37
19 Experimental results of the hybrid technique with the Big Trees configuration on the Australian dataset (testing)	38
20 Experimental results of the hybrid technique with the Small Trees configuration on the Australian dataset (training).....	39
21 Experimental results of the hybrid technique with the Small Trees configuration on the Australian dataset (testing)	40

List of Tables

1 Duplication frequency for replication of data classification rules produced by the ACO technique as initial populations of the GP technique	18
2 Performance comparison of classification rules extracted by the redesigned ACO technique and AntMiner+	25
3 Summary of results from all techniques during training.....	40
4 Summary of results from all techniques during testing	40

1. Introduction

Genetic Programming (GP), introduced by John Koza (1992), is one of the most popular evolutionary computation techniques. Based on Darwin's principle of natural selection, GP is a fast and efficient method for searching a large space of possible solutions in the selected problem with the goal of finding one of the best solutions. The initial input for GP consists of randomly created generation of solutions that are represented by tree structures. A fitness function is used to evaluate these solutions and the genetic operations, crossover and mutation, are applied to them to create the next generation of solutions. The genetic operations are repeatedly applied on successive generations of solutions, which keep evolving until: (a) the solutions pass a predetermined threshold on quality as determined by the fitness function or (b) the number of iterations reaches a predefined number (even though an acceptable solution may not yet be found). Thus, the evolution of solutions can be computationally expensive.

Ant Colony Optimization (ACO) was proposed by Marc Dorigo (1992) as a meta-heuristic method mimicking the behavior of real ants. It uses the indirect communication taking place among ants when travel to food sources and then carry food back to their colony. In particular, ants leave pheromone trails, which allow other ants to retrace their paths. When other ants detect the existing pheromone trails, it affects their decisions on choosing paths as they are attracted more to paths with higher levels of pheromone. Pheromone levels increase on paths that have been followed by more ants recently, attracting even more ants. In the end, all ants converge on to one specific path, which represents a solution achieved in a heuristic fashion. In essence, the ACO-based technique performs a parallel search for a solution over several constructive

computational threads and selects the best solution from the collective results derived from the interaction between different search threads.

This thesis aims to investigate whether a new hybrid method combining ACO and GP will yield better performance than the original GP method. The credit scoring problem was used as an application to verify the hypothesis: *a hybrid GP-ACO method is better than either GP or ACO alone*. Specifically, a hybrid model based on GP and ACO was used for classifying credit applicants as good or bad risk. In particular, after obtaining the initial result from the ACO method, we used the result as an initial population for GP with the goal of finding a better solution faster than the original GP (i.e., better fitness with fewer generations).

To aid GP in its search for solutions, this thesis proposes a new hybrid technique that aims to improve GP performance by providing it with ACO generated candidate solutions instead of randomly generated solutions as the initial population. This idea is based upon the assumption that the ACO approach views the data classification problems differently from the GP technique and it is also capable of learning from its application on the targeted dataset. This is because the ACO approach uses heuristically cooperative parallel searches in each convergence, whereas the GP technique relies on heuristically independent parallel searches in each generation. Thus, the proposed hybrid technique will enable the GP technique to leverage the insights gained by the ACO technique, improving its capability in generating good solutions.

An investigation of the proposed hybrid technique was carried out through the implementation of GP, ACO, and hybrid approaches in the C programming language. These codes were then compiled using GNU C compiler version 4.4 and executed on a laptop computer running Ubuntu Linux operating system version 11.10. For evaluation, two well-known datasets

used for credit scoring problems, referred to as the Australian and German datasets, were deployed. These datasets were obtained from the Machine Learning Depository at the University of California, Irvine. The experiments were conducted by executing GP, ACO, and hybrid codes to produce classification rules for these two datasets. These rules were then evaluated for their data classification accuracies.

During the training, the experimental results show that the rules produced by the GP technique yield an average of 74.73 percent accuracy on the German dataset and an average of 88.3 percent accuracy on the Australian dataset, whereas the rules produced by the ACO technique yield an average of 97.6 percent accuracy on the German dataset and an average of 56.87 percent accuracy on the Australian dataset. The rules produced by the hybrid technique yield an average of 82.27 percent accuracy on the German dataset and an average of 92.67 percent on the Australian dataset.

During the testing, the experimental results show that the rules produced by the GP technique yield an average of 68.7 percent accuracy on the German dataset and an average of 84.3 percent accuracy on the Australian dataset, whereas the rules produced by the ACO technique yield an average of 97.6 percent accuracy on the German dataset and an average of 56.9 percent accuracy on the Australian dataset. The rules produced by the hybrid technique yield an average of 70.7 percent accuracy on the German dataset and an average of 84.3 percent on the Australian dataset.

In summary, the data classification rules from the proposed hybrid technique consistently outperform the rules from the GP technique in terms of data classification accuracy during the training, demonstrating that the proposed hybrid technique is indeed capable of improving the

GP technique in its search for good solutions even though these same rules from both techniques on average yield similar performance in testing.

This thesis is organized into five chapters. After a brief overview in this introduction chapter, the related work is surveyed in Chapter 2. The third chapter explains the proposed hybrid technique. Chapter 4 presents and discusses the experimental results. Finally, the thesis is concluded in Chapter 5.

2. Related Work

A survey of published research work that applies ACO, GP, or a hybrid of both techniques to a variety of interesting problems is presented below. It shows that the proposed work is not only in an active and competitive research area with several practical applications, but also a unique way to hybridize the ACO and GP techniques when compared with the published ACO-GP hybrid techniques in the literature.

Lahsasna et al. (2010) published a survey of several research studies on credit scoring models. These studies can be classified into groups based on their adopted methodologies such as statistical models, artificial neural networks (ANN), and evolutionary computation.

Statistical methods can be applied to customer records for calculating credit scores and classifying them as good or bad risk (Hand & Henley, 1997; Thomas et al., 2002). Another group of researchers in this area adopted ANN as an individual method or in combination with other methods. In the study by West (2000), five different techniques in artificial intelligence were examined with the accuracy of their respective results compared with one another. Other researchers have shown that the ANN-based models produce better results than statistical models (Desai et al., 1996; Baesens et al., 2003; Malhotra & Malhotra, 2003). Although ANNs were successfully applied to different problems, because of their black box nature, their underlying logic that determines a prediction cannot be observed. This is considered as a shortcoming of ANN-based methodologies (Vellido et al., 1999). Finally, the last group of researchers used evolutionary computation methods. Hoffmann et al. (2002) compared two fuzzy classifiers,

namely genetic fuzzy and neuro-fuzzy, for credit scoring. The results demonstrated that the genetic fuzzy classifier outperformed the neuro-fuzzy classifier.

Li et al. (2011) introduced a multi-criteria model for credit decision making. They adopted a two-stage iterated algorithm, which helps improved classification since there are different criteria for optimization. In their study, the evolutionary strategies-based model is proposed for optimizing parameters based on the concept of natural evolution. The results demonstrated that their proposed model is suitable for credit classification, and the evolution strategy is a viable technique for optimizing the parameters.

Ong et al. (2005) used GP to create a credit scoring model and they compared results from the model with results from multi-layer perceptron, classification and regression tree (CART), C4.5, rough sets, and logistic regression (LR) models. Their empirical results showed that GP performed better than other models in terms of accuracy of credit scoring and flexibility of credit score models.

Abdou (2009) created a credit scoring model for Egyptian banks and used statistical, GP, and other techniques to predict consumer loan quality. The results show that GP-based model is the most accurate.

Martens et al. (2010) used ACO to predict credit rating of customers in various datasets. We are particularly interested in their application of ACO on the German dataset to extract credit evaluation rules. They used only 5 of the 20 attributes in the German dataset to predict the credit rating and their results show that ACO produces the rules that yield 71.9% prediction accuracy.

Parpinelli et al. (2002) proposed the ACO-based data miner for inducing rules from data. The classification rule is created using an if-statement with multiple conditions. These conditions have multiple terms which are AND together. Each term contains attribute, operator, and attribute's value. The operator element is always the equal (==) operator. The partial construction of rule follows the path that ant chooses. The choice of term to be added is related to the amount of pheromone and the heuristic function. The goal is covering more cases in the dataset, following the rule antecedents. The authors compare the performance of Ant Miner and C4.5 algorithm using six different datasets from various application domains. Their results show that Ant Miner produces results with higher accuracy than C4.5 algorithm on four datasets. They are tied in term of accuracy on one of the datasets and C4.5 algorithm beats Ant Miner on prediction accuracy in one of the datasets.

Liu et al. (2004) introduced an improved version of Ant Miner algorithm called Ant Miner 3. They enhanced the existing algorithm by encouraging the ants from having a bias toward exploration. The likelihood for the ants to choose terms that have not been used in previously constructed rules is increased by adding a transition rule. The proposed algorithm was evaluated using breast cancer and tic-tac-toe datasets from UCI ML dataset repository. They conclude that their version of Ant Miner provides a balance between exploitation and exploration.

The Ant Miner algorithm is also applied to Web content mining problem, which is associated with a large number of attributes. The goal of such research is discovering a good set of classification rules to group Web pages according to their subjects. The result of the study by Holden and Freitas (2004) shows the result of classification using Ant Miner algorithm yields

better or at least the same performance when compare to data mining using methods such as C4.5 and C2. In addition, it produces much simpler rules than the data mining algorithms.

Another important research topic in swarm intelligence for rule induction is in the exploration of search space and in maintaining a balance between exploration and exploitation. Galea (2002) investigated the role of exploration in her master thesis. She concluded that increasing the ant population size and using different transition rule affect the results. The experiments show the potential of proposed method for increasing the prediction accuracy. However, these changes can also decrease and increase computation time when applied.

Swaminathan (2006) presented a hybrid approach combining a statistical technique called C4.5 algorithm with Ant Miner algorithm. This proposed approach is an attempt to represent the numerical ranges for numerical attributes in the Ant Miner algorithm in such a way that they are equivalent to the nominal attributes. He applied probability density function for continuous attributes in Ant Minder algorithm. The rule induction process in this research work began with the discretization of the values of numerical attribute by using C4.5 algorithm to find the range in the continuous domain. By using C4.5 to pre-process the datasets, the author was effective narrow the scope of the conditions used for establishing the rules. This type of dataset pre-processing allows the proposed approach to outperform two previous versions of the Ant Miner algorithm.

White and Salehi-Abari (2008) proposed an ACO-inspired crossover operation for GP. The main idea of their approach adapts the pheromone concepts from the ACO methods for computing the components of fitness function in the evolutionary process of GP. This approach

to fitness function influences the selection of sub-trees for crossover operations, and thus, in my opinion, it also guides GP to produce different solutions. Their results based on the evaluation of symbolic regression problem show that GP produces the solution faster (i.e., in less number of generations) in several cases.

Runka (2009) applied the natural evolutionary mechanism of Genetic Programming (GP) to build an improved formula for an ACO algorithm. The proposed method was evaluated using the TSP problem. The author obtained the new formula for calculating the probability that ants will choose a particular path between a pair of cities in the TSP problem using GP. This new formula replaced the original ACO formula. Specifically, the new formula created from each GP individual expression tree is equivalent to the numerator of the original formula while the denominator is obtained through the summation of GP individuals for all possible destination cities. For the experiments, the author used the strongly-typed GP with both crossover and mutation operators enabled. Once the new GP-based formula is obtained and deployed to calculate the probability for a path between each pair of cities in the ACO algorithm, each ant picks the next city based on one of the three selection methods:

- Roulette selection
- Greedy Selection
- Tournament Selection

The performance obtained from ACO algorithm with this new GP-based formula is used for calculating the fitness score of that individual. Since different ACO methods for selecting ants'

next city directly influence the outcome, it also influences the fitness score used by GP to produce the individual in the subsequent generation. Their result with Roulette selection shows an improvement over the original ant system. However, the author stated that further testing on a variety of problems must be done before a declaration of the proposed technique as a replacement for the original formula can be made.

3. Proposed Hybrid Technique

The proposed hybrid technique is explained in this chapter. It is a hybrid of the Ant Colony Optimization (ACO) technique and the Genetic Programming (GP) technique. To apply the ACO technique for hybridization with the GP technique efficiently and effectively, the formula for calculating heuristic values and the selection logic for a decision on the pheromone accumulation in the ACO technique were redesigned. Specifically, the first component in the redesigned ACO technique is the network model of the credit scoring problem that serves as an input for the ACO technique. This network model is based on a dataset of credit history records from the targeted financial institution. Each record in this dataset contains multiple attributes reflecting the credit history of a particular customer.

Generally, attributes in the dataset are categorized into two groups: nominal (or qualitative) and continuous (or numerical). While both types of attributes have assigned values, the value of a nominal variable is considered as a label rather than a number. It is also called a categorical variable. For example, an attribute on housing status is considered a nominal type with value of 0 for own and 1 for rent. On the other hand, the value of a continuous variable is considered as numeric, such as 6, 14, or 1,362, and the relative magnitude of its value is important. For example, an attribute on the level of annual incomes is considered continuous, where the value of \$200,000 indicates twice the magnitude of the value of \$100,000.

We designed the following network diagram (see Figure 1) to represent the attributes from the sample dataset. In this diagram, a group of vertices is defined for each attribute in the dataset. Each vertex represents a possible value for a particular attribute. For example, attribute G_1

represents checking account status of a customer. The four possible values of this attribute are: (0) for customer with checking account balance of less than zero, (1) for customer with checking account balance of zero to less than 200, (2) for customer with checking account balance of 200 or higher, and (3) for customer with no checking account. Each of these four possible values is represented as a node in the diagram.

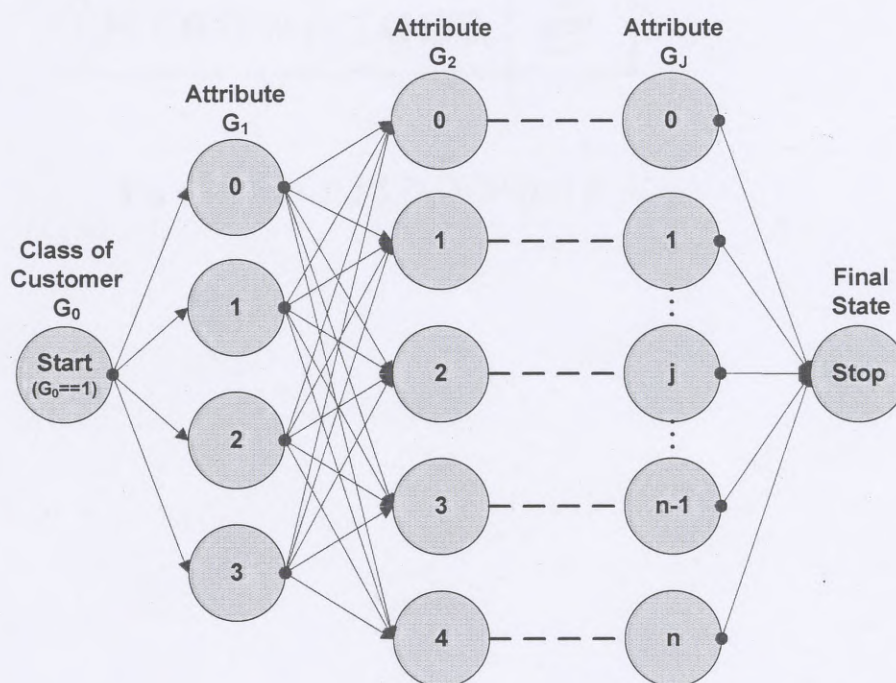


Figure 1. ACO network diagram with attributes from the sample dataset (possible values for G_j are $\{0, \dots, j, \dots, n\}$)

Under the ACO technique, a group of ants begins moving from the start vertex to the stop vertex. The start vertex represents a desired group of customers with good credit rating. At the end, each route from the start vertex to the stop vertex selected by an ant constitutes a rule for classification of creditworthiness. Each ant randomly selects a path between two attributes based

on the probability value assigned to that path at that time. The probability value is defined and calculated as follows.

$$P_{(I,i,J,j)}(t) = [\tau_{(I,i,J,j)}(t)]^\alpha [\eta_{J,j}(t)]^\beta / \sum_{(k=0 \text{ to } n)} [\tau_{(I,i,J,k)}(t)]^\alpha [\eta_{J,k}(t)]^\beta, \text{ where}$$

- $\tau_{(I,i,J,j)}$ is the amount of pheromone associated with the rule that represents a path between attributes I and J with their values corresponding to indexes i and j,
- $\eta_{J,j}$ is the heuristic value of node J with value j,
- k represents all of the possible values for attribute J,
- n is the maximum possible value of attribute J, and
- α and β are the relative weights or influences of the pheromone level and the heuristic value, respectively.

The probability (P) used by an ant to make a decision in selecting a particular path between attributes I and J with their values corresponding to indexes i and j is calculated according to the formula above. We derived this formula from the original ACO-based technique proposed by Dorigo (1992). Our modifications to the original formula were made to accommodate the characteristics of credit scoring problem as well as the attributes of the dataset used in this study.

The pheromone level (τ) is calculated with the following formula.

$\tau_{(I,i,J,j)}(t=0) = 1 / \text{num}(\text{links between all vertices})$, where $\text{num}()$ is the total number of paths between vertices according to network diagram, and

$$\tau_{(I,i,J,j)}(t+1) = \rho (\tau_{(I,i,J,j)}(t)) + Q_{\text{best}}, \text{ where}$$

- ρ is the pheromone evaporation rate with value between 0 and 1, and

- Q_{best} is the sum of Confidence and Coverage values for the Rule R, which are calculated by the following formulas.

$$\text{Confidence} = \text{num}(R_{\text{antec}} == \text{true} \&\& R_{\text{consq}} == \text{true}) / \text{num}(R_{\text{antec}} == \text{true}),$$

$$\text{Coverage} = \text{num}(R_{\text{antec}} == \text{true} \&\& R_{\text{consq}} == \text{true}) / N, \text{ where}$$

- $R_{\text{antecedent}}$: $G_J == j \&\& \dots \&\& G_M == m \&\& \dots \&\& G_A == a$
- $R_{\text{consequent}}$: $G_0 == 1$ (i.e., Class of Customer is Good)
- $\text{num}(R_{\text{antec}} == \text{true} \&\& R_{\text{consq}} == \text{true})$ represents the number of records in which rule antecedent and rule consequent are true,
- $\text{num}(R_{\text{antec}} == \text{true})$ is the number of records in which rule antecedent is true, and
- N is the current number of records in the dataset.

Finally, the heuristic value (η) is derived from the following formula where the Correctness is multiplied with the percentage of records that an attribute with specific value is covered.

$$\eta_{j,j} = W * \text{Correctness},$$

$$\text{Correctness} = \text{num}(G_J == j \&\& G_0 == 1) / \text{num}(G_J == j), \text{ and}$$

$$W = \text{num}(G_J == j \&\& G_0 == 1) / \sum_{(k=0 \text{ to } n)} \text{num}(G_J == k \&\& G_0 == 1)$$

- $\text{num}(G_J == j \&\& G_0 == 1)$ represents the number of records in which the J^{th} attribute has the value j and that the very same records are classified class of customer as good, and
- $\text{num}(G_J == j)$ is the number of records in which the J^{th} attribute has the value j .

From these formulas, we initiate the ants by setting the initial pheromone level on each link according to the formula for τ with $t=0$ and by calculating the initial heuristic value of each node according to the formula for computing η value.

Then, the probability for each link is calculated using the formula for computing P value. Each ant selects one of the links from a vertex in the attribute G_0 called "Class of Customer" (see Figure 1) to one of the vertices in the attribute G_1 using the computed P values. This step is repeated to select one of the links between two attributes. Once a complete path from the start vertex (Class of Customer) to stop vertex (Final State) is obtained, the pheromone level of the links on the selected path is updated using the formula for updating the value of τ with $t \neq 0$, which applies both pheromone evaporation and accumulation, whereas the pheromone level of other links in the network is updated only through pheromone evaporation.

Then, the probability for each link (P value) is re-calculated and the steps above are repeated until the pheromone levels of updated τ on all links on a particular path is greater than τ_{\max} and all other links has the updated pheromone level of τ_{\min} . In this study, we set τ_{\max} equal to the initial pheromone level calculated with τ formula where $t=0$, whereas τ_{\min} is set to zero.

A rule based on the path with all links with τ greater than τ_{\max} is extracted and the rows of data records that are covered by this newly obtained rule are deleted from the dataset. To achieve better validation rate, the obtained rules are pruned by ranking the Confidence rate among the selected attributes. As an example, the ACO code may extract a rule consisting of nine attributes, but it yields a very low Confidence rate due to over fitting. By ranking the Confidence rate of these attributes individually, we can select a subset of attributes that provides a good Confidence rate. In this study, we set a threshold for a good Confidence rate at 50 percent.

Next, the pheromone level on each link is re-initialized using the formula τ with $t=0$, and the heuristic value of each vertex is re-calculated and the whole process repeated over the remaining rows of data records in the dataset. The subsequent rules obtained from ACO are then combined with the rules obtained earlier. Finally, the ACO code is terminated when all of the records in the training data are covered, when the addition of newly extracted rules is repeated three times without any improvement on the accumulative accuracy, or when the ACO codes have extracted 18 rules, whichever occurs first.

Once the data classification rules are obtained from the ACO technique, they were used as the initial populations in the first generation of the GP evolutionary process. To understand this hybridization, the GP evolutionary process is explained below.

GP uses evolution to find good solutions to the given problem. The initial population in GP consists of programs that are represented by parse trees containing functions and terminal sets as nodes. These parse trees are generated randomly as the initial populations in the first generation of the evolutionary process under the original GP. Specially, attributes from the dataset specification are used for defining the terminal sets, which are the leaf nodes of the tree. Data values of these leaf nodes are typically derived from external inputs, functions with no arguments, and predefined constants. The function sets consist of arithmetic functions (such as addition, subtraction, multiplication, and division), comparative functions (such as equal-to, greater-than, and less-than), and Boolean functions (such as AND, OR, NAND, NOR, XOR, and NOT) for manipulating the attributes from the terminal sets. An example of a GP sub-tree is shown below in Figure 2. The leaf nodes are attribute A1 and a constant integer value of 1000.

According to an operator in their parental node, this sub-tree returns a multiplication result of these two leaf nodes.

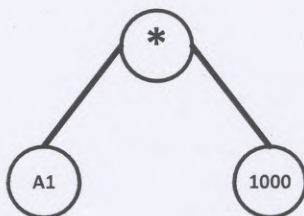


Figure 2: A sample sub-tree

In this study of the hybrid technique, the randomly generated initial populations are replaced by the initial populations derived from the data classification rules generated by the ACO technique. This replacement was carried out in two different ways under the experiments conducted in this study. The first method generates a single parse tree from all data classification rules created by the ACO technique. This parse tree construction incorporates the priority level of each data classification rule through the duplication frequency (see Table 1). For example, the first data classification rule created by the ACO technique has the highest priority, and thus, it is copied many more times in the single parse tree than the last data classification rules created by the ACO technique. Then, this single parse tree (named Big Trees) is duplicated to match with the initial population size for the GP technique. The second method (so named Small Trees) generates a parse tree from each data classification rule created by the ACO technique. The priority level of each data classification rule is incorporated through duplication frequency (again, see Table 1) when each of these small parse trees is copied to match with the initial population size for the GP technique.

Table 1. Duplication frequency for replication of data classification rules produced by the ACO technique as initial populations of the GP technique (N is the number of rules extracted by the ACO technique)

Priority Level	Duplication Frequency
1	N
2	N-1
3	N-2
4	N-3
5	N-4
:	:
N	1

The distance between an ideal solution and the proposed solution from the current generation was used in this study to evaluate the fitness function. This fitness function is important because it influences the search effectiveness. As an example, the fitness function can be defined as shown in equation below, where *abs* is the absolute function, FF_g defines the fitness function of each of GP's solutions, E_i represents the expected result for each record in the dataset, O_i represents the output value from the GP's solution for each record in the dataset, and N is the total number of records in the dataset:

$$FF_g = \frac{\sum_{i=1}^N abs(E_i - O_i)}{N}$$

Thus, this equation calculates the fitness of an individual solution from GP by averaging the distances between all of the expected results and the real outputs. A smaller gap between the expected value and the output value means better fitness. Once the fitness function is defined, the genetic operations are applied to the initial population. There are two important genetic operators in GP, namely recombination (or crossover) and mutation.

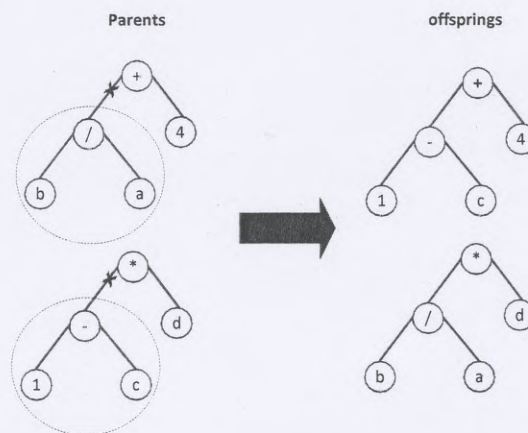


Figure 3: An example of crossover operation

Recombination operator (or sub-trees crossover) randomly selects sub-trees, and then, swaps them at the crossover point to produce new trees, which are called off-springs. An example of a crossover operation is shown in Figure 3. In this figure, the left sub-trees (highlighted in dotted circles) are exchanged at the crossover point.

There are several possible mutation operations as follows: (i) sub-tree mutation, which picks a random sub-tree and replaces it with a randomly generated sub-tree, (ii) replace a random leaf node with a sub-tree, (iii) randomly swap sub-trees of non-leaf node, (iv) randomly change an ephemeral random constant (ERC), and (v) select two disjoint sub-trees and swap them. In this

study, the sub-trees are derived from defined attributes given in the dataset. Then, the off-springs from genetic operations become the population of the next generation and the process is repeated until the terminating condition, such as the maximum number of generations or the threshold for acceptable solutions, is met. In this study, the maximum number of generations was set at 100 and the threshold for acceptable solutions was set at 100 percent accuracy. The experimental results are presented and discussed in the next chapter.

4. Results and Discussions

To evaluate the effectiveness of the proposed hybrid technique, the Australian and German datasets for credit scoring problem were deployed. These datasets are available from the Machine Learning Repository at the University of California, Irvine. The chapter presents and discusses the results collected from the redesigned ACO, the original GP, and the proposed hybrid techniques on both datasets.

4.1 The Redesigned ACO Results

This section presents results collected from the redesigned ACO technique for creditworthiness classification. From a total of 1,000 records in the German dataset, we used the first 500 records for training and extracting rules, and the remaining 500 records for testing them.

The ACO code was executed with different configurations to find the threshold on the number of attributes in each rule that still produced acceptable results. The results show the maximum number of attributes in the rule that has an acceptable level of Confidence is 4 attributes. These four attributes have been selected dynamically among those with highest Confidence rate. The experiment with this configuration was repeated three times and the accumulative accuracy from the obtained rules in each experiment is shown in Figure 4 below. These graphs represent the accumulative accuracy after each convergence that introduces and combines the extracted rules with the existing ones. The accumulative accuracy rate starts at 50 percent in all three experiments when the first rule is extracted by ACO and increases by 15.6 percent to 65.6 percent after the total of 18 rules are extracted and combined together using the OR Boolean operators.

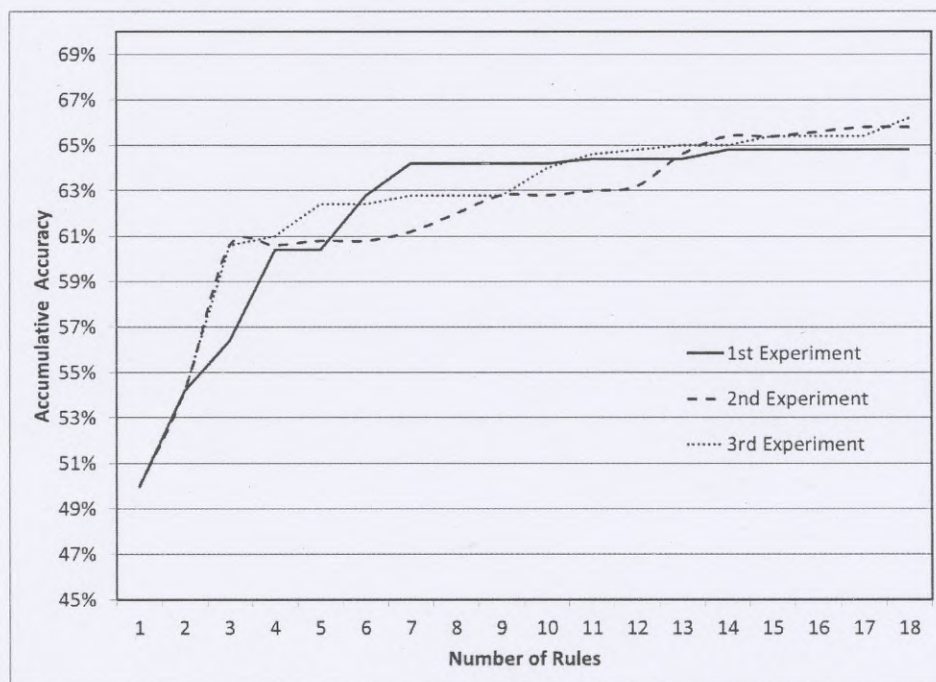


Figure 4. Accumulative accuracy of ACO-based rules during training for three experiments on the German dataset (4 attributes)

Additionally, we also collected experimental results for rules with only 3 attributes. The accumulative accuracy of these rules from obtained during each of the three experiments is shown in Figure 5 below. It shows that the accumulative accuracy is increased after each rule is added to the rule(s) extracted earlier at the end of each convergence. On average, the accuracy starts from 66.2 percent and peaks at 97.4 percent after all of the 18 rules from 18 convergences are combined during the training. The result shows the accumulative accuracy of the best rules with 3 attributes to be 98.6 percent, which is 33 percent greater than the accumulative accuracy of the best rules with 4 attributes.

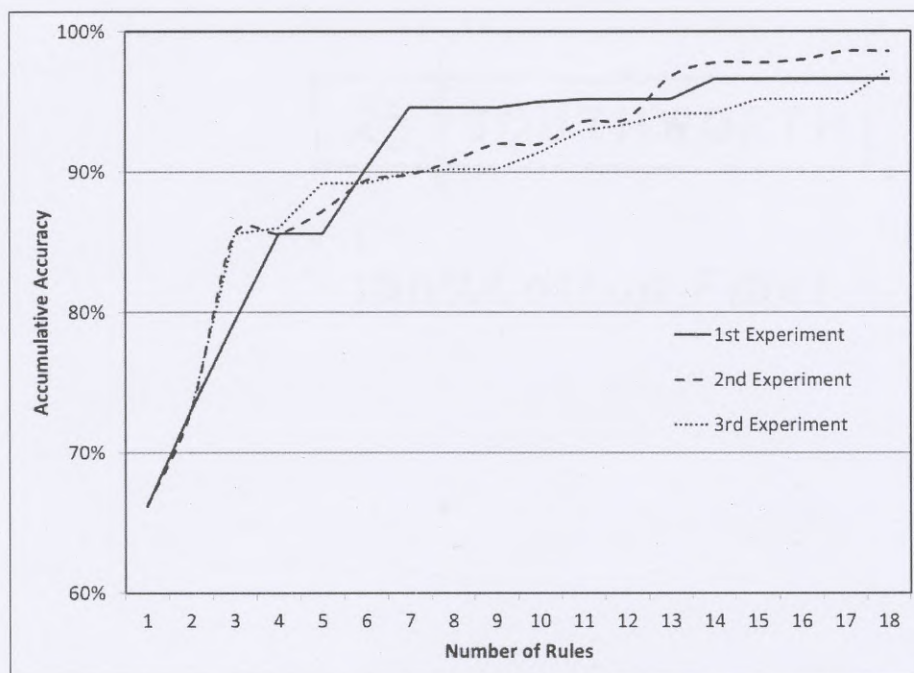


Figure 5. Accumulative accuracy of ACO-based rules during training for three experiments on the German dataset (3 attributes)

To demonstrate the accuracy of ACO-based rules obtained during the training, we tested them on the remaining 500 records of the German dataset. The best test results among all three experiments obtained using 3- and 4-attribute rules are shown in Figure 6 below. The highest accumulative accuracy found for 3-attribute rules was for the second experiment. At 98.6 percent, this accuracy is 28 percent higher than accumulative accuracy of the best result obtained from 4-attribute rules that produces 70.6 percent accumulative accuracy.

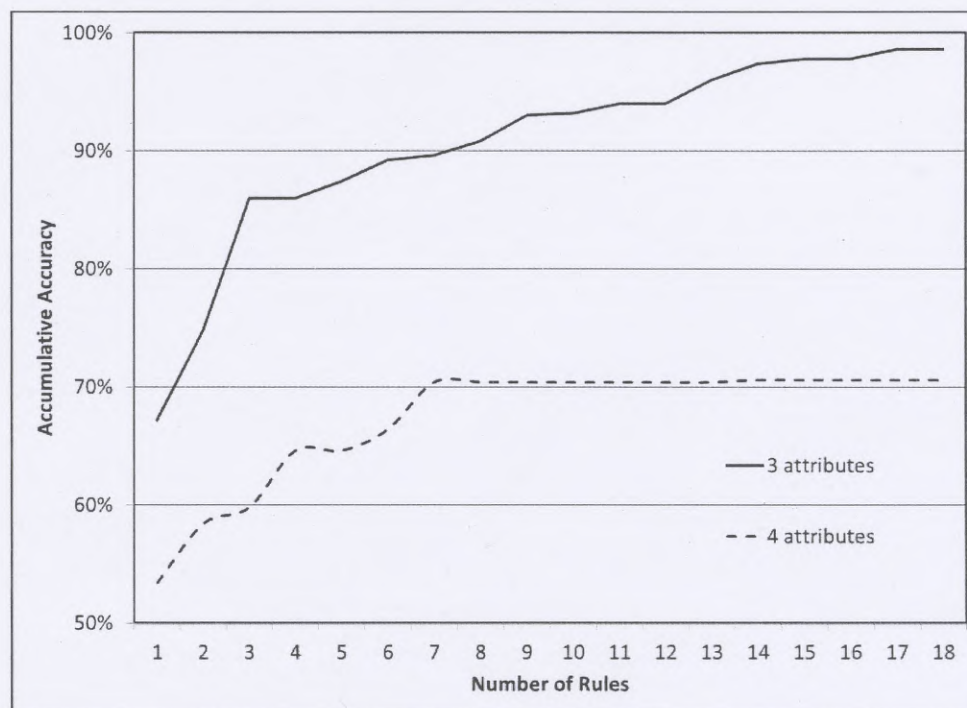


Figure 6. Accumulative accuracy of ACO-based rules on the German dataset during testing

Finally, we compared the performance of classification rules extracted by the redesigned ACO technique against the rules obtained from AntMiner+, the best known implementation of ACO-based technique for credit scoring. The experiment was carried out using the German dataset with the first 500 records used for training and the remaining 500 records for testing. The results are shown in Table 2 below.

According to the results shown in Table 2 below, we can observe the following trends: (a) the rules extracted by the redesigned ACO technique outperform rules extracted by AntMiner+, (b) the redesigned ACO technique can extract more rules than AntMiner+, and (c) The redesigned ACO technique produces rules with higher quality than AntMiner+. Specifically, the final classification rule from the redesigned ACO technique outperforms rules from AntMiner+

by 26.8 percent. AntMiner+ terminates with only 4 classification rules because the additional rule it produced leads to lower accuracy rate, whereas the redesigned ACO technique produces 18 rules that always led to higher accuracy rates.

Table 2. Performance comparison of classification rules extracted by the redesigned ACO technique and AntMiner+

Algorithm	Number of Rules	Number of Operators	Number of Attributes	Accumulative Accuracy (training)	Accumulative Accuracy (testing)
Rules Extracted by our ACO-based Technique	1	5	3	66.2%	67.2%
	2	11	4	73.0%	74.8%
	3	17	4	85.6%	86.0%
	4	23	4	85.6%	86.0%
	5	29	5	87.2%	87.4%
	6	35	5	89.4%	89.2%
	7	41	6	89.8%	89.6%
	8	47	6	90.8%	90.8%
	9	53	6	92.0%	93.0%
	10	59	7	92.0%	93.2%
	11	65	7	93.6%	94.0%
	12	71	8	93.8%	94.0%
	13	77	8	96.8%	96.0%
	14	83	8	97.8%	97.4%
	15	89	8	97.8%	97.8%
	16	95	8	98.0%	97.8%
	17	101	8	98.6%	98.6%
	18	107	8	98.6%	98.6%
AntMiner+	4	29	5	69.4%	71.8%

Finally, when comparing rules that have the same number of operators and attributes as the AntMiner+'s rules, our rules outperformed those of the AntMiner+ by more than 15 percent (as shown in Figure 7 below). Thus, classification rules extracted by the redesigned ACO technique have both higher accuracy and efficiency than the rules produced by the best known ACO-based technique for credit scoring problems.

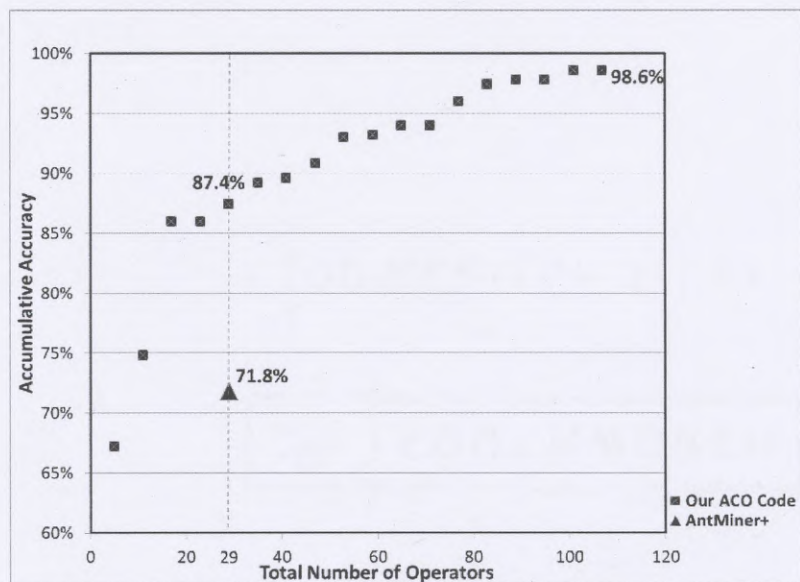


Figure 7. Comparative performance of classification rules extracted by our ACO-based technique against AntMiner+ based on the total number of operators

In another experiment that used the Australian dataset containing a total of 690 records, the first 345 records were used for training and extracting data classification rules, and the remaining 345 records were used for testing them. Unlike the German dataset, there are no descriptions on any of the attributes in the Australian dataset. The results show the data classification rules generated by the redesigned ACO yield an average of 56.87 percent accuracy during the training and an average of 56.9 percent accuracy during the testing (see Figures 8 and 9 below). This result on the Australian dataset is attributed to a very different correlation between the number of attributes, the number of records, and the bias on classification of credit applicants when compared with the German dataset. Specifically, despite the blind nature of the Australian dataset, it is known that the number of attributes in the Australian dataset is 30 percent fewer, the

number of records in the Australian dataset is 30 percent less, and the number of credit applicants classified as bad credit risk is almost double when compared with the German dataset.

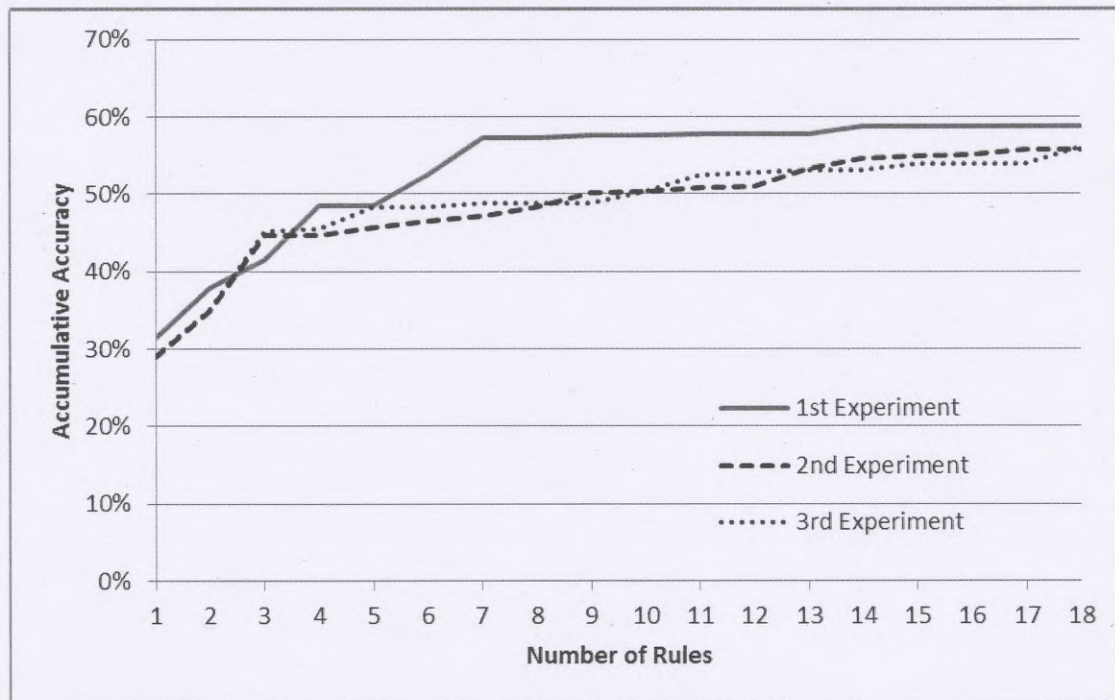


Figure 8. Accumulative accuracy of ACO-based rules during training for three experiments on the Australian dataset

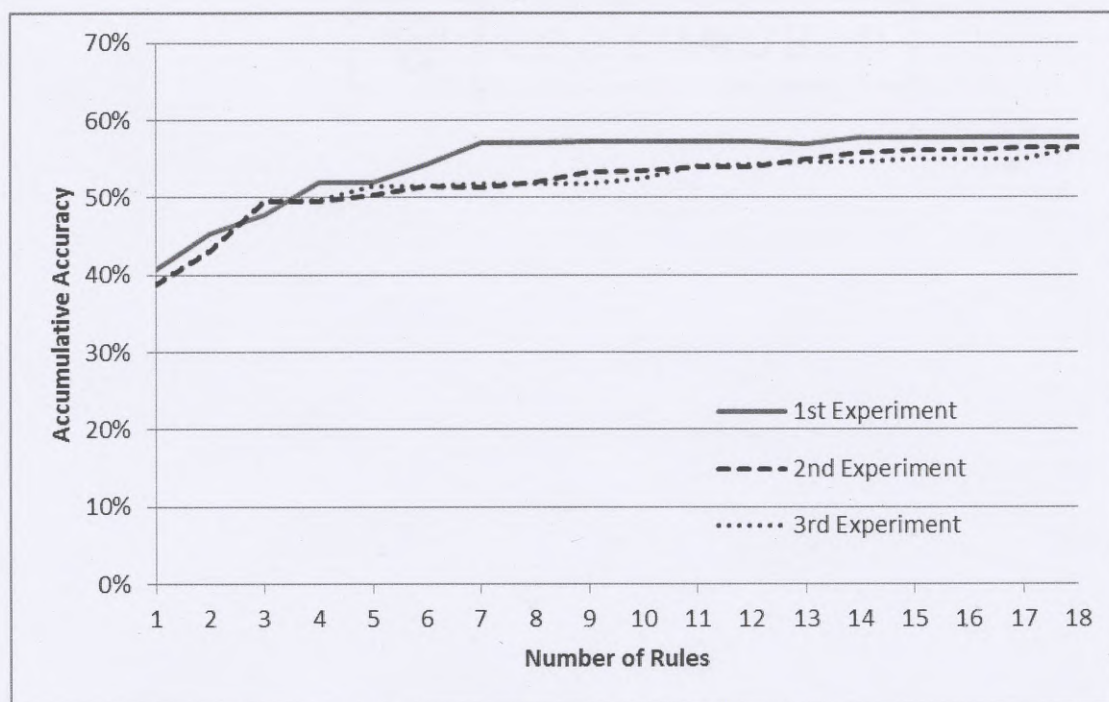


Figure 9. Accumulative accuracy of ACO-based rules during testing for three experiments on the Australian dataset

In summary, these mixed results from the rules generated by the ACO technique on two different datasets are reflections of the heuristic nature of data classification techniques based on the swarm intelligence methodology.

4.2 The GP Results

This section presents results collected from the GP technique for creditworthiness classification. From a total of 1,000 records in the German dataset, we used the first 500 records for training and extracting rules, and the remaining 500 records for testing them. The experiment was repeated three times. As shown in Figure 10 below, during the training, the results show that the rules produced from the GP technique yield a classification accuracy of 74.6 percent, 74.8 percent, and 74.8 percent, in the first, second, and third experiment, respectively. Thus, during

the training, the data classification rules produced by the GP technique yield an average of 74.73 percent accuracy overall on the German dataset.

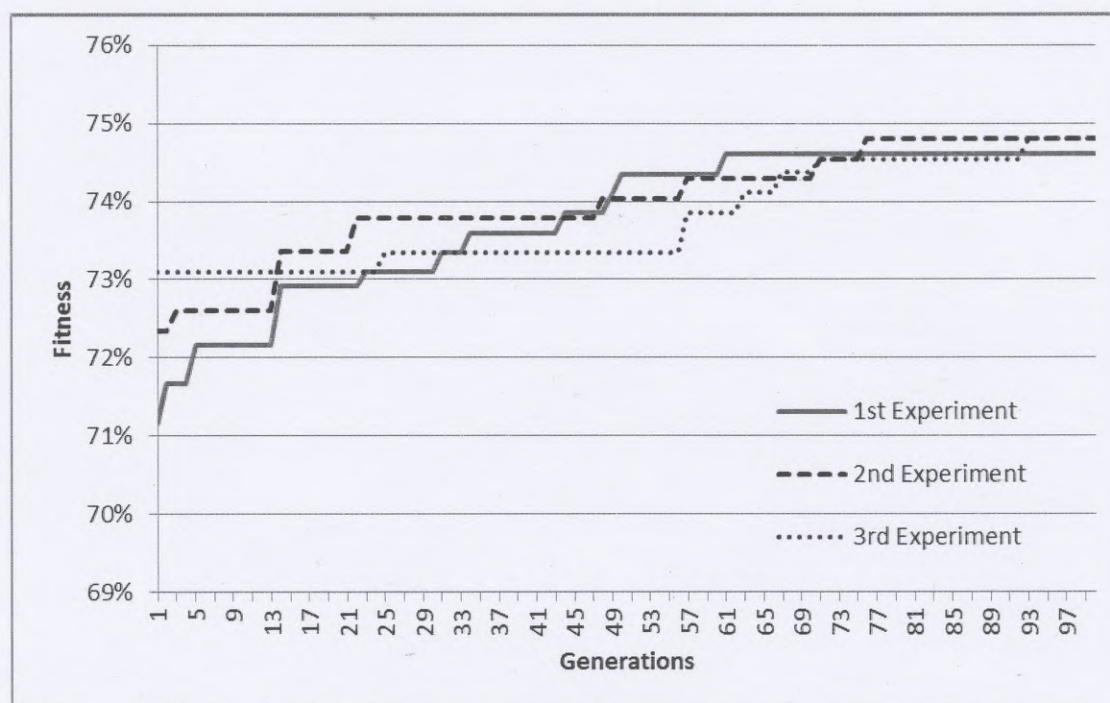


Figure 10. Experimental results of the GP technique on the German dataset (training)

As shown in Figure 11, during the testing, the results show that the rules produced from the GP technique yield a classification accuracy of 67.8 percent, 69.4 percent, and 69 percent, in the first, second, and third experiment, respectively. Thus, during the testing, the data classification rules produced by the GP technique yield an average of 68.7 percent accuracy overall on the German dataset.

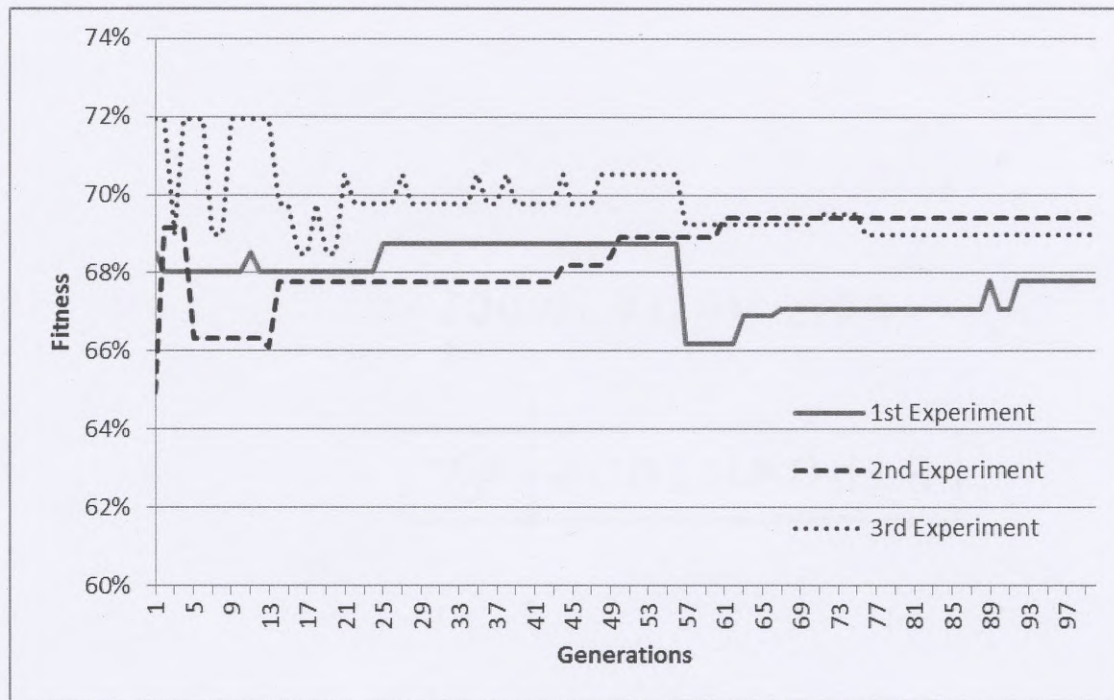


Figure 11. Experimental results of the GP technique on the German dataset (testing)

In another experiment, the Australian dataset was deployed. From a total of 690 records in the dataset, the first 345 records were used for training and extracting data classification rules, and the remaining 345 records were used for testing them. During the training, the results from three experiments using the same configuration show the fitness for the first generation in all experiments is around 85 percent (see Figure 12). It increases progressively over the subsequent generations and reaches 89, 88, and 87 percent fitness level in the 61st, 77th, and 93rd generations in the first, second and third experiments, respectively. Thus, during the training, the data classification rules generated by the GP technique have an average of 88.3 percent accuracy on the Australian dataset.

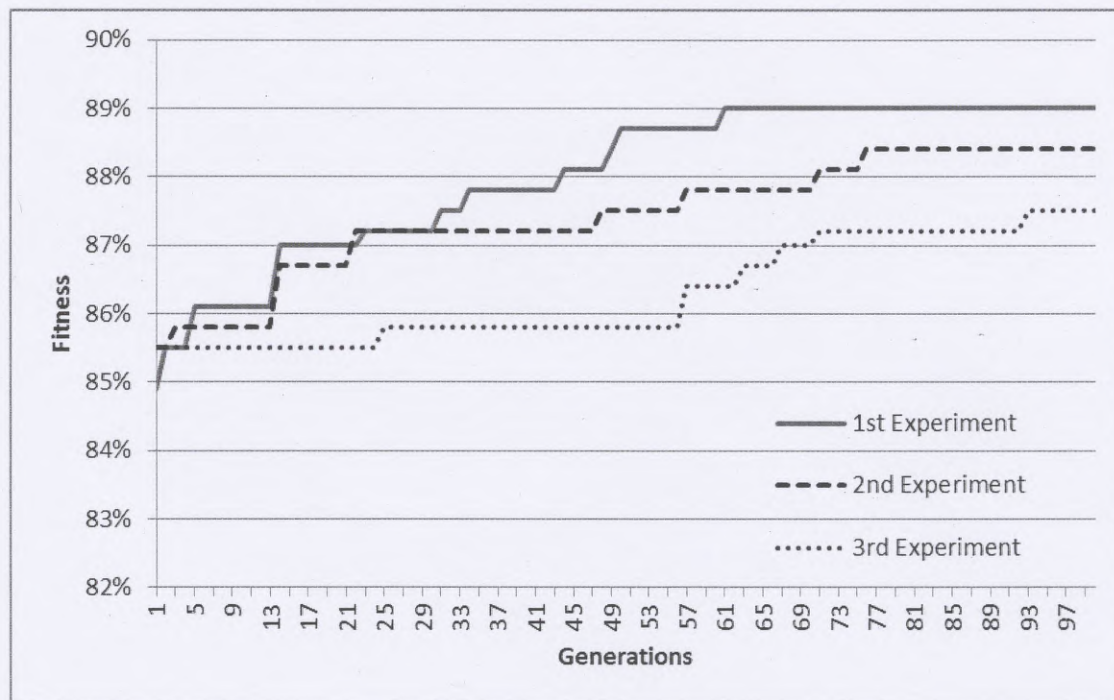


Figure 12. Experimental results of the GP technique on the Australian dataset (training)

As shown in Figure 13, during the testing, the results from three experiments using the same configuration show 85.2, 85.8, and 82 percent fitness levels in the first, second and third experiments, respectively. Thus, during the testing, the data classification rules generated by the GP technique have an average of 84.3 percent accuracy on the Australian dataset.

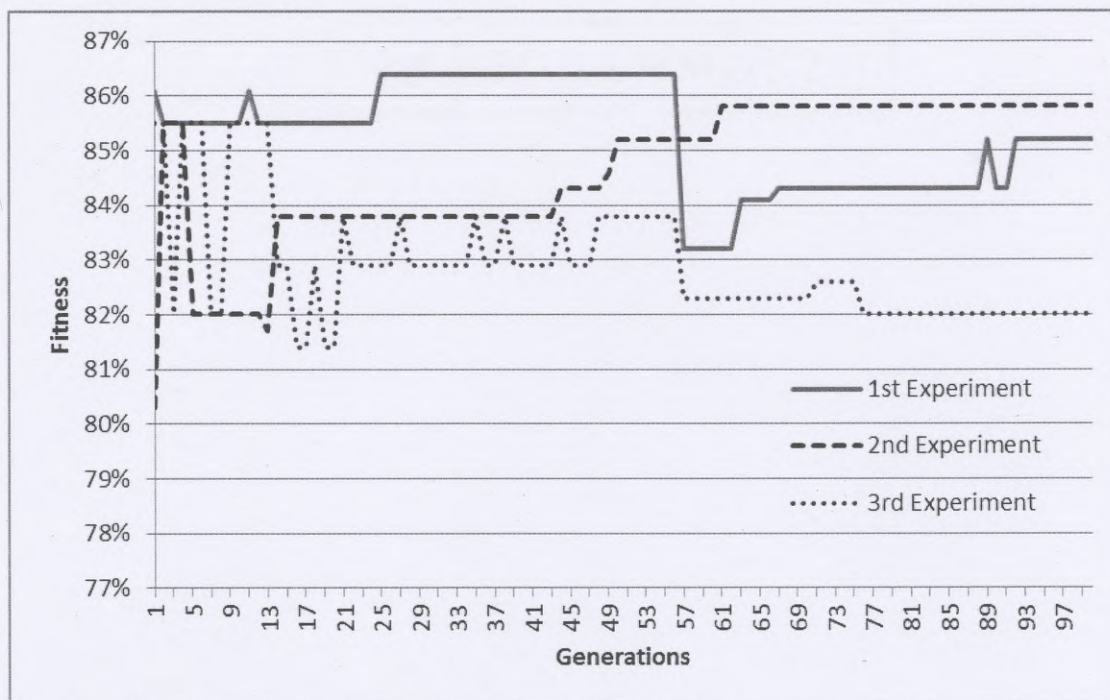


Figure 13. Experimental results of the GP technique on the Australian dataset (testing)

4.3 The Hybrid Technique Results

The evaluation of the proposed hybrid technique was carried out on both the German and Australian datasets. There are two hybrid configurations called Big Trees and Small Trees (see an explanation page 16 of Chapter 3). The experiments were repeated three times under each configuration. The experiments on the German dataset using hybrid technique with the Big Trees configuration yield the results shown in Figures 14 and 15 below.

From a total of 1,000 records in the German dataset, we used the first 500 records for training and extracting rules, and the remaining 500 records for testing them. During the training, the results show that the rules produced from the hybrid technique with the Big Trees configuration yield a classification accuracy of 81.4 percent, 82.2 percent, and 80.3 percent, in the first,

second, and third experiment, respectively. Thus, during the training, the data classification rules produced by the hybrid technique with the Big Trees configuration yield an average of 81.3 percent accuracy overall on the German dataset.

During the testing, the results show that the rules produced from the hybrid technique with the Big Trees configuration yield a classification accuracy of 70.8 percent, 71.2 percent, and 70.2 percent, in the first, second, and third experiment, respectively. Thus, during the testing, the data classification rules produced by the hybrid technique with the Big Trees configuration yield an average of 70.7 percent accuracy overall on the German dataset.

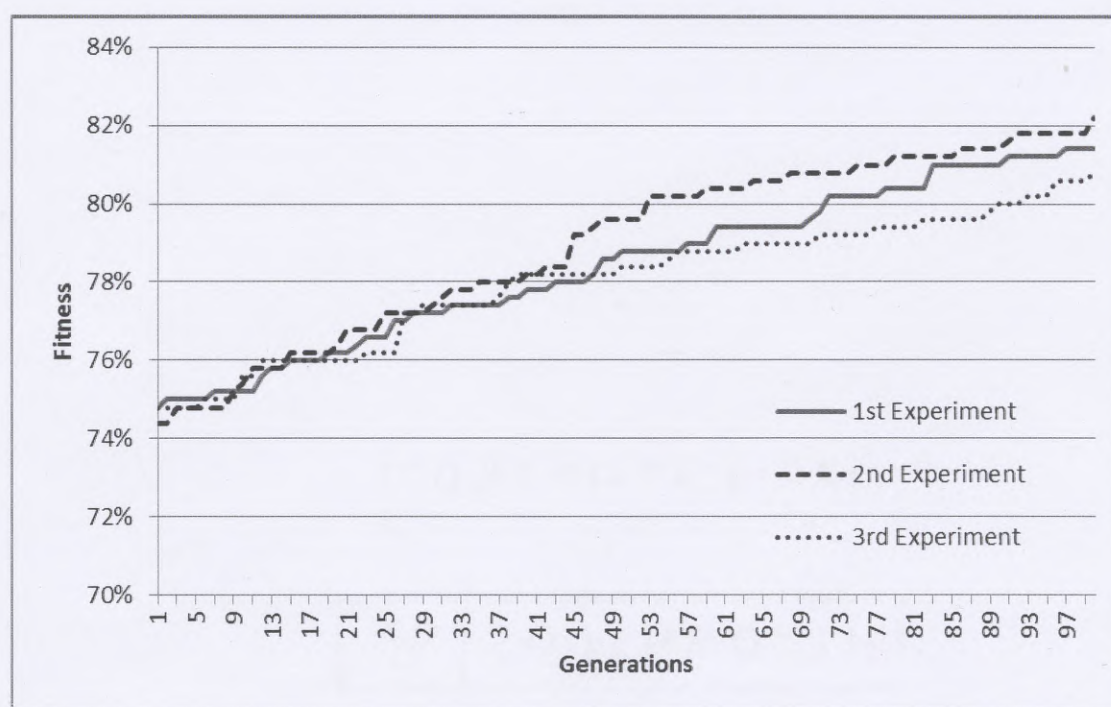


Figure 14. Experimental results of the hybrid technique with the Big Trees configuration on the German dataset (training)

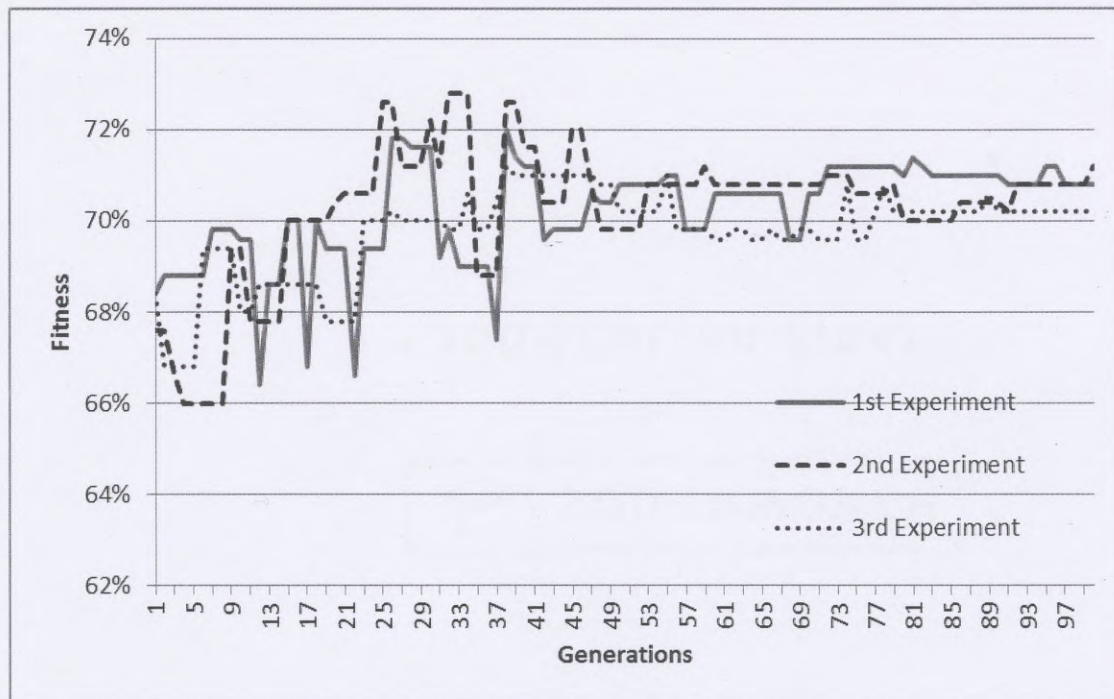


Figure 15. Experimental results of the hybrid technique with the Big Trees configuration on the German dataset (testing)

The experiments on the German dataset were repeated three additional times with the Small Trees configuration. The results are shown in Figure 16 and 17 below. During the training, the results show that the rules produced from the hybrid technique with the Small Trees configuration yield a classification accuracy of 81.8 percent, 83 percent, and 82 percent, in the first, second, and third experiment, respectively. Thus, during the training, the data classification rules produced by the hybrid technique with the Small Trees configuration yield an average of 82.27 percent accuracy overall on the German dataset.

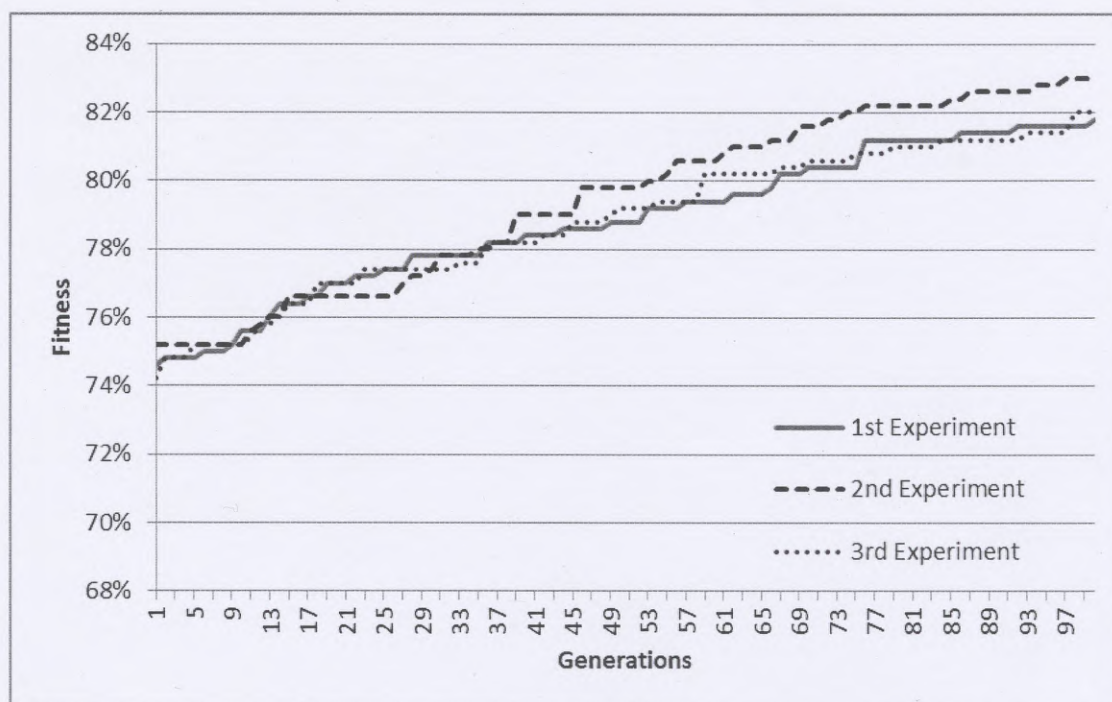


Figure 16. Experimental results of the hybrid technique with the Small Trees configuration on the German dataset (training)

During the testing, the results show that the rules produced from the hybrid technique with the Small Trees configuration yield a classification accuracy of 71.2 percent, 68.8 percent, and 70.4 percent, in the first, second, and third experiment, respectively. Thus, during the testing, the data classification rules produced by the hybrid technique with the Small Trees configuration yield an average of 70.1 percent accuracy overall on the German dataset.

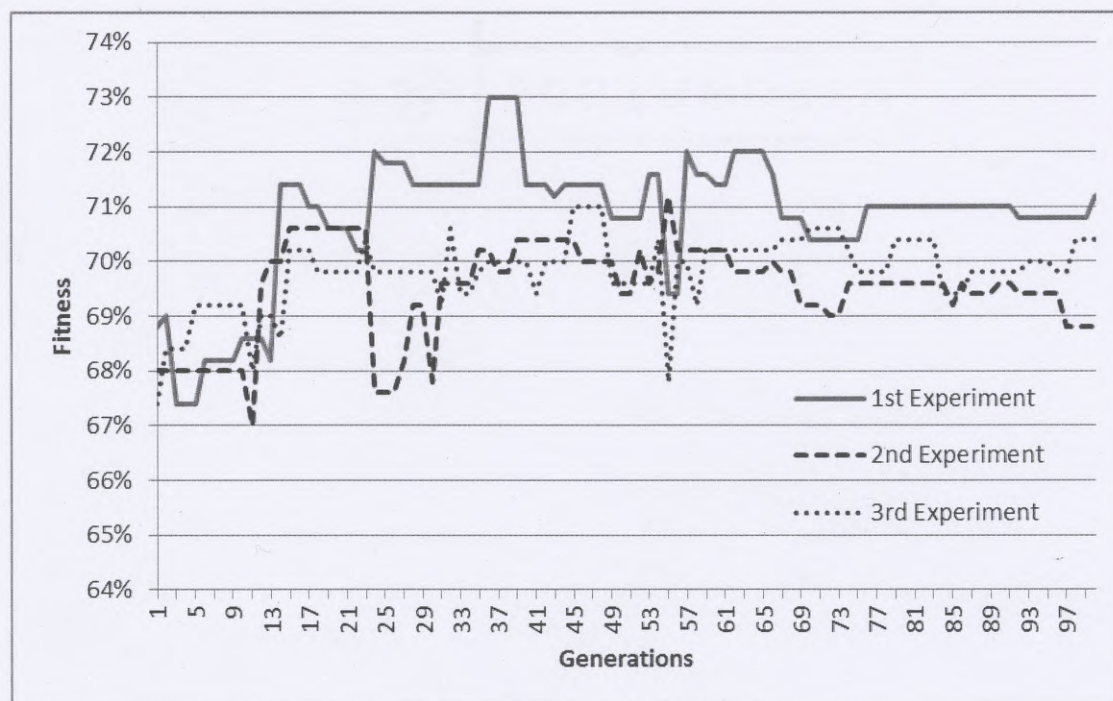


Figure 17. Experimental results of the hybrid technique with the Small Trees configuration on the German dataset (testing)

The experiments on the Australian dataset using hybrid technique with the Big Trees configuration yield the results as shown in Figures 18 and 19.

From a total of 690 records in the Australian dataset, we used the first 345 records for training and extracting rules, and the remaining 345 records for testing them. During the training, the results show that the rules produced from the hybrid technique with the Big Trees configuration yield a classification accuracy of 93.3 percent, 91.9 percent, and 92.8 percent, in the first, second, and third experiment, respectively. Thus, during the training, the data classification rules produced by the hybrid technique with the Big Trees configuration yield an average of 92.67 percent accuracy overall on the Australian dataset.

During the testing, the results show that the rules produced from the hybrid technique with the Big Trees configuration yield a classification accuracy of 83.5 percent, 84.9 percent, and 84.6 percent, in the first, second, and third experiment, respectively. Thus, during the testing, the data classification rules produced by the hybrid technique with the Big Trees configuration yield an average of 84.3 percent accuracy overall on the Australian dataset.

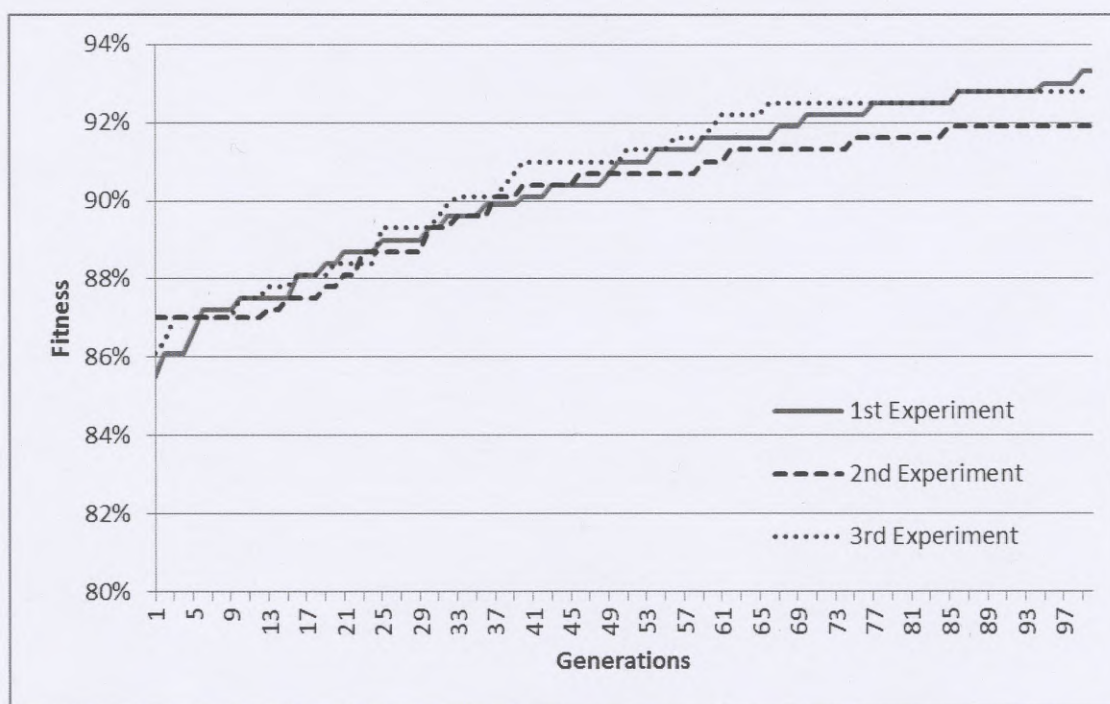


Figure 18. Experimental results of the hybrid technique with the Big Trees configuration on the Australian dataset (training)

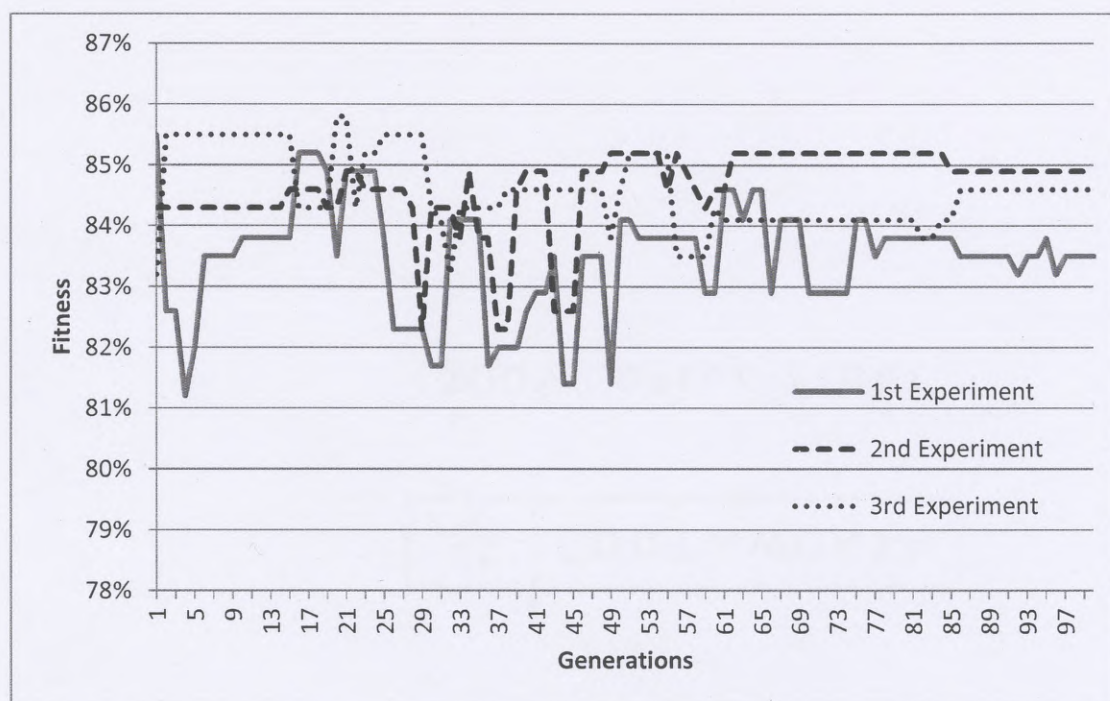


Figure 19. Experimental results of the hybrid technique with the Big Trees configuration on the Australian dataset (testing)

The experiments on the Australian dataset were also repeated three additional times with the Small Trees configuration. The results are shown in Figures 20 and 21 below. During the training, the results show that the rules produced from the hybrid technique with the Small Trees configuration yield a classification accuracy of 92.5 percent, 91.9 percent, and 91.3 percent, in the first, second, and third experiment, respectively. Thus, during the training, the data classification rules produced by the hybrid technique with the Small Trees configuration yield an average of 91.9 percent accuracy overall on the Australian dataset.

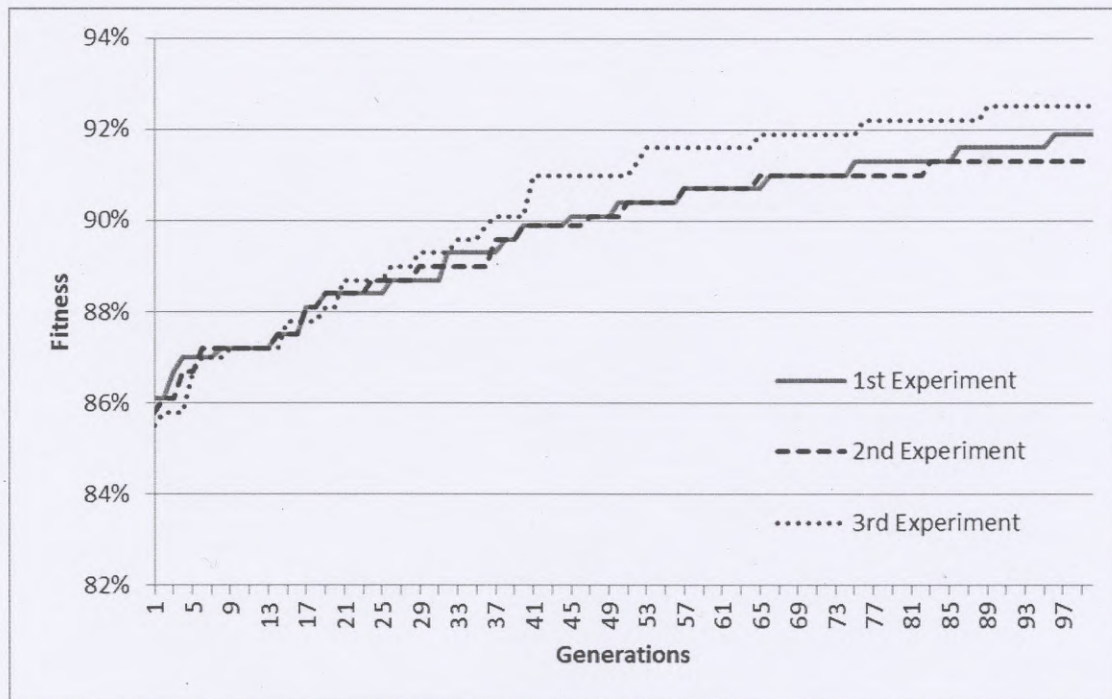


Figure 20. Experimental results of the hybrid technique with the Small Trees configuration on the Australian dataset (training)

During the testing, the results show that the rules produced from the hybrid technique with the Small Trees configuration yield a classification accuracy of 84.1 percent, 83.8 percent, and 84.9 percent, in the first, second, and third experiment, respectively. Thus, during the testing, the data classification rules produced by the hybrid technique with the Small Trees configuration yield an average of 84.3 percent accuracy overall on the Australian dataset.

In summary, the experimental results shown in Tables 3 and 4 below demonstrate that the proposed hybrid technique is indeed capable of improving on the GP technique in its search for good solutions during training although all tree techniques yield similar accuracy during testing.

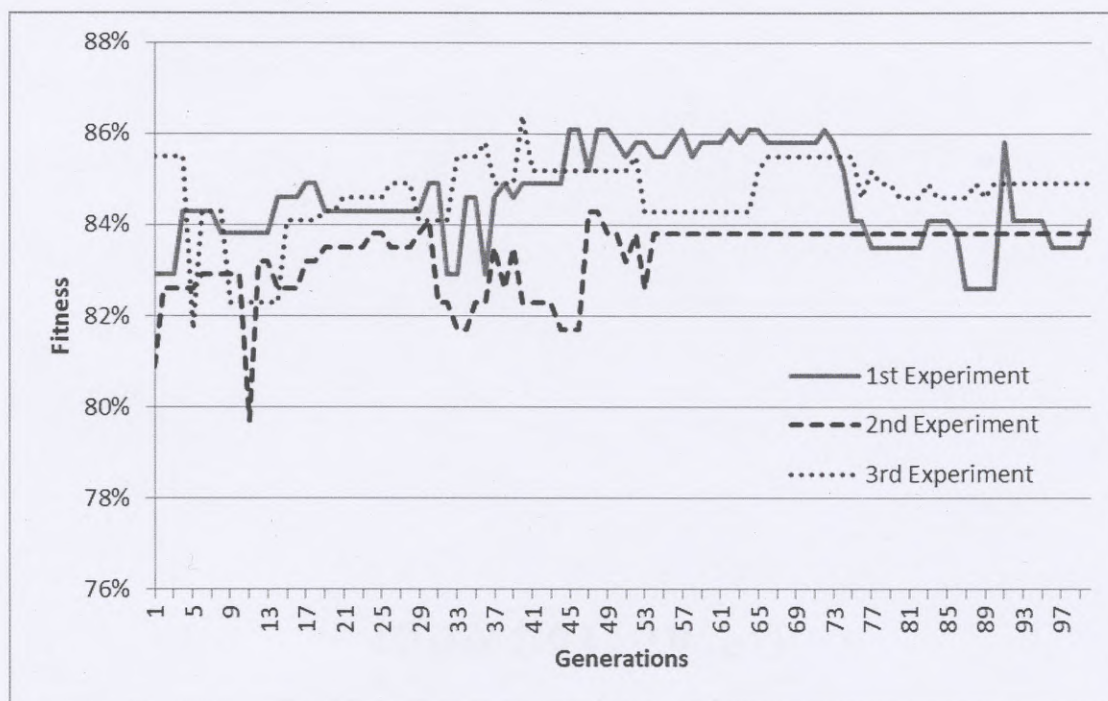


Figure 21. Experimental results of the hybrid technique with the Small Trees configuration on the Australian dataset (testing)

Table 3. Summary of results from all techniques during training

Dataset	ACO (percent)	GP (percent)	Hybrid (percent)
German	98.6, 96.6, 97.2	74.6, 74.8, 74.8	81.8, 83, 82
	Average: 97.4	Average: 74.7	Average: 82.3
Australian	58.8, 55.7, 56.1	89.2, 88.4, 87.5	93.3, 91.9, 92.8
	Average: 56.9	Average: 88.3	Average: 92.7
Average	77.2	81.5	87.5

Table 4. Summary of results from all techniques during testing

Dataset	ACO (percent)	GP (percent)	Hybrid (percent)
German	97.2, 98.6, 97.8	67.8, 69.4, 69	70.8, 71.2, 70.2
	Average: 97.8	Average: 68.7	Average: 70.7
Australian	57.7, 56.5, 56.5	85.2, 85.8, 82	83.5, 84.9, 84.6
	Average: 56.9	Average: 84.3	Average: 84.3
Average	77.25	76.5	77.5

5. Conclusions and Future Work

This thesis proposed a new hybrid technique consisting of Genetic Programming (GP) and Ant Colony Optimization (ACO) for improving on GP performance in finding good solutions. The main idea of this hybrid technique is leveraging the outputs and insights from the ACO technique as the initial population of solutions for the GP technique, giving the GP technique a head start. The research work in this study was carried out through the implementation of ACO, GP, and the hybrid techniques. This was followed by the evaluation of their performance in terms of generating good data classification rules using the two well-known datasets for the credit scoring problem. These datasets, referred to as the Australian dataset and the German dataset, were obtained from the Machine Learning Repository at the University of California, Irvine.

During training, the experimental results show that the rules produced by the GP technique yield an average of 74.73 percent accuracy on the German dataset and an average of 88.3 percent accuracy on the Australian dataset, whereas the rules produced by the ACO technique yield an average of 97.6 percent accuracy on the German dataset and an average of 56.87 percent accuracy on the Australian dataset. The rules produced by the hybrid technique yield an average of 82.27 percent accuracy on the German dataset and an average of 92.67 percent on the Australian dataset. Thus, the data classification rules from the proposed hybrid technique consistently outperform the rules from the GP technique in terms of data classification accuracy, demonstrating that the proposed hybrid technique is indeed capable of improving the GP technique in its search for good solutions during the training even though the rules from all techniques yield virtually the same results on average in testing.

Although a part of experimental results from research work in this thesis has already been published (Aliehyaei & Khan, 2011; Aliehyaei, 2012), there are several possibilities for future work. The most interesting direction is applying the proposed technique on the credit history of consumers in the United States, making an impact in the real world. However, this would require a significant collaboration and cooperation from the three major credit bureaus, namely Equifax, Experian, and Transunion. Another possibility for future work is applying the proposed technique on different datasets from a variety of interesting problems in the industries, such as healthcare, biomedical, and games, among others. Finally, it would also be interesting to explore other evolutionary computation and swarm intelligence techniques on these problems.

References

- Abdou, H. (2009). Genetic Programming for Credit Scoring: The Case of Egyptian Public Sector Banks. *Expert System with Application*, 36, 11402-11417.
- Aliehyaei, R. (2012). Efficient and Effective Classification of Creditworthiness using Ant Colony Optimization. *ACM Southeast Conference*, 83-88.
- Aliehyaei, R. & Khan, S. (2011). A Credit Scoring Model based on Strongly-typed Genetic Programming. *IASTED Computational Intelligence and Bioinformatics Conference*, 16-23.
- Baesens, B., Setiono, R., Mues, C., & Vanthienen, J. (2003). Using Neural Network Rule Extraction and Decision Tables for Credit Risk Evaluation. *Computer Journal of Management Science*, 49(3), 312-329.
- Desai, V., Crook, J., & Overstreet, G. (1996). A Comparison of Neural Networks and Linear Scoring Models in the Credit Union Environment. *European Journal of Operational Research*, 95(6), 24-37.
- Dorigo, M. (1992). Optimization, Learning, and Natural Algorithms (Doctoral Dissertation). Diploma Elettronica e Informazione, Politecnico di Milano, Italy.
- Galea, M. (2002). Apply Swarm Intelligence. Master Thesis, University of Edinburgh, 2002.
- Hand, J., & Henley, W. (1997). Statistical Classification Methods in Consumer Credit Scoring. *Computer Journal of the Royal Statistical Society Series a Statistics in Society*, 160(3), 523-541.

- Hoffmann, F., Baesens, B., Martens, J., Fput, F., & Vanthienen, J. (2002). Comparing a Genetic Fuzzy and a Neurofuzzy Classifier for Credit Scoring. *International Computer Journal of Intelligent Systems*, 17(11), 1067-1083.
- Holden, N. & Freitas, A. (2004). Web Page Classification with an Ant Colony Algorithm.
- Koza, J. (1992). *Genetic Programming*. MIT Press, Cambridge, Massachusetts.
- Li, J., Wei, L., Li, G., & Xu, W. (2011). An Evolution Strategy-based Multiple Kernels Multi-criteria Programming Approach: The Case of Credit Decision Making. *Decision Support Systems*, 51(2), 292-298.
- Liu, B., Abbass, H., & McKay, B. (2004). Classification Rule Discovery with Ant Colony Optimization. *IEEE Computational Intelligence Bulletin*, (3)1, 31-35.
- Malhotra, R., & Malhotra, K. (2003). Evaluating Consumer Loans Using Neural Networks. *Omega Computer Journal*, 31(2), 83-96.
- Martens, D., Gestel, T., Backer, M., Haesen, R., Vanthienen, J., & Baesens, B. (2010). Credit Rating Prediction Using Ant Colony Optimization, Catholic University of Leuven.
- Ong, C., Huang, J., & Tzeng, G. (2005). Building Credit Scoring Models using Genetic Programming. *Computer Journal of Expert Systems with Applications*, 29(2), 41-47.
- Parpinelli, R., Lopes, H., & Freitas, A. (2002). Data Mining with an Ant Colony Optimization Algorithm. *IEEE Transactions on Evolutionary Computation*, 6, 321-322.

- Runka, A. (2009). Evolving an Edge Selection Formula for Ant Colony Optimization. The Genetic and Evolutionary Computation Conference (GECCO), 1075-1082.
- Swaminathan, S. (2006). Rule Induction using Ant Colony Optimization for Mixed Variable Attributes. Master Thesis, Texas Institute of Technology.
- Thomas, L. C., Edelman, D. B., & Crook, J.N. (2002). Credit Scoring and its Applications. SIAM Monographs on Mathematical Modeling and Computation, Philadelphia, PA.
- Vellido, A., Lisboa, G., & Vaughan, J. (1999). Neural Networks in Business: A Survey of Applications. Proceedings of Expert Systems with Applications, 51-70.
- West, D. (2000). Neural Network Credit Scoring Models. Computers and Operations Research, 27(3), 1131-1152.
- White, T. & Salehi-Abari, A. (2008). A Swarm-based Crossover Operator for Genetic Programming. The Genetic and Evolutionary Computation Conference (GECCO), 1345-1346.

